# The trace norm constrained matrix-variate Gaussian process for multitask bipartite ranking

Oluwasanmi Koyejo, Cheng Lee, and Joydeep Ghosh, *Fellow, IEEE*

**Abstract**—We propose a novel hierarchical model for multitask bipartite ranking. The proposed approach combines a matrix-variate Gaussian process with a generative model for task-wise bipartite ranking. In addition, we employ a novel trace constrained variational inference approach to impose low rank structure on the posterior matrix-variate Gaussian process. The resulting posterior covariance function is derived in closed form, and the posterior mean function is the solution to a matrix-variate regression with a novel spectral elastic net regularizer. Further, we show that variational inference for the trace constrained matrix-variate Gaussian process combined with maximum likelihood parameter estimation for the bipartite ranking model is jointly convex.

Our motivating application is the prioritization of candidate disease genes. The goal of this task is to aid the identification of unobserved associations between human genes and diseases using a small set of observed associations as well as kernels induced by gene-gene interaction networks and disease ontologies. Our experimental results illustrate the performance of the proposed model on real world datasets. Moreover, we find that the resulting low rank solution improves the computational scalability of training and testing as compared to baseline models.

**Index Terms**—Gaussian process, Multitask learning, Bipartite ranking, Trace norm.



## 1 Introduction

Ranking is the task of learning an ordering for a set of items. In bipartite ranking, these items are drawn from two sets, known as the positive set and the negative set. Bipartite ranking involves learning an ordering that ranks the positive items ahead of the negative items [1], [2], [3], [4]. This paper proposes a generative model for bipartite ranking and an extension of bipartite ranking to the multitask domain. Our approach combines a latent multitask regression function with task-wise ordered observation variables. We employ a non-parametric matrix-variate Gaussian process prior for the multitask regression. Further, we propose a novel trace constrained variational inference approach that imposes useful low rank structure on the multitask regression.

Multitask learning (MTL) exploits inter-task relationships to improve the prediction quality over single task learning [5], [6]. An important class of methods in this domain are based on the matrix-variate Gaussian process (MV-GP) and closely related models for vector valued reproducing kernel Hilbert space (RKHS) function estimation [7]. The MV-GP is an extension of the matrix-variate Gaussian distribution [8] to (possibly) infinite dimensional feature spaces. Alternatively, the MV-GP may be understood as an extension of the scalar valued Gaussian process [9] to vector valued responses. The MV-GP is a useful model for learning multiple correlated tasks, as it jointly models the correlations across examples, and across tasks. The MV-GP has been applied to link analysis, transfer learning [10], collaborative prediction [11] and multitask learning [12] among other applications.

Our motivating application is the prioritization of disease genes. Genes are segments of DNA that determine specific characteristics; over 20,000 genes have been identified in humans, which interact to regulate various functions in the body. Researchers have identified thousands of diseases, including various cancers and respiratory diseases such as asthma [13], caused by mutations in these genes. The standard approach for discovering disease-causing genes are genetic association studies [14]. However, these studies are often tedious and expensive to conduct. Hence, computational methods that can reduce the search space by predicting a prioritized list of candidate genes for a given disease are of significant scientific interest.

The disease-gene prioritization task has received a significant amount of study in recent years [15], [16], [17], [18]. The task is challenging because all the observed responses correspond to known associations and the states of the unobserved associations are unknown, i.e., there are no reliable negative examples. Such problems are also known *single class* or *positive-unlabeled* (PU) learning tasks [19]. A common approach for this task is to learn a model that that maximizes the classification accuracy between the positive class and the unlabeled class [20]. In the collaborative filtering literature, such single class tasks have also been addressed using the low rank matrix factorization approach [21].

Recent work suggests that a model trained to rank the positive class ahead of the unknowns can be effective for ranking the unknown positive items ahead of unknown

• *O. Koyejo and J. Ghosh are with the Department of Electrical and Computer Engineering, and C. Lee is with the Department of Biomedical Engineering, University of Texas at Austin, Austin, TX, 78712. E-mail: {sanmi.k@, chlee@, ghosh@ece}.utexas.edu*

negative items [19]. Further, the scientific use case for gene prioritization depends on (and is evaluated by) the accuracy of the ranked list produced [15], [17]. For these reasons, disease-gene prioritization is well posed as a bipartite ranking task. A low rank model induces significant correlation between the predictions of different tasks. This assumption matches observations made by domain experts that the gene ranking profiles of diseases are exhibit strong correlations [18], [22]. This assumption is further validated by the empirical performance of the low rank model.

Low rank structure is a typical constraint in several real world multitask and matrix learning problems. However, despite its use for multitask learning, the standard MV-GP does not model low rank structure. Further, memory requirements for the MV-GP scale quadratically with data size, and naïve computation scales cubically with data size. These computational properties limit the applicability of the MV-GP to large scale problems. The hierarchical factor Gaussian process (factor GP) has been proposed as an alternative for problems with low rank structure [23], [24]. Here, latent row and column factors are drawn from a Gaussian process prior. The result is a model with mean of user-selected rank. We argue that the factor GP is an unsatisfactory model for two reasons: (i) the resulting posterior mean function is the solution of a non-convex optimization problem, and (ii) the posterior covariance is often intractable. We will show that the proposed trace constrained MV-GP provides the same low rank structure benefits without the drawbacks of the factor GP model. The proposed variational inference is jointly convex in the mean and the covariance, and the posterior covariance is given in closed form.

As a computational model, the optimization problem for the mean function of the trace constrained MV-GP is equivalent to kernel multitask learning with the sum of squared errors cost function [25] combined with a novel regularizer. We will show that this regularizer can be expressed as a weighted sum of the Hilbert and the trace norms. We call the resulting regularization the *spectral elastic net*, highlighting its relationship to elastic net regularization for variable selection in finite dimensional linear models [26]. To the best of our knowledge, ours is the first application of the spectral elastic net regularizer to matrix estimation and kernel multitask learning.

This paper proposes a novel generative model for multitask bipartite ranking and a novel constrained variational inference approach for the matrix variate Gaussian process applied to the disease-gene prioritization task. The main contributions of this paper are as follows:

- We propose a novel variational inference approach for matrix-variate Gaussian process regression using a trace norm constraint (section 3). This constraint typically results in a regression matrix of low rank.
- We propose a novel generative model for bipartite ranking (section 4). To our knowledge, ours is the first such generative model proposed in the literature.
- We show that variational inference for the latent regression model combined with maximum likelihood parameter estimation for the bipartite ranking is jointly convex (section 4.3).
- We evaluate the proposed model empirically and show that it outperforms the state of the art domain specific model for the disease-gene prioritization task (section 5).

**The Kronecker product and the vec operator:** We will make significant use of the Kronecker product and the vec$(\cdot)$ operator. Given a matrix $\mathbf{A} \in \mathbb{R}^{P \times Q}$, vec$(\mathbf{A}) \in \mathbb{R}^{PQ}$ is the vector obtained by concatenating columns of $\mathbf{A}$. Given matrices $\mathbf{A} \in \mathbb{R}^{P \times Q}$ and $\mathbf{B} \in \mathbb{R}^{P' \times Q'}$, the Kronecker product of $\mathbf{A}$ and $\mathbf{B}$ is denoted as $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{PP' \times QQ'}$. A useful property of the Kronecker product is the identity: vec$(\mathbf{AXB}) = (\mathbf{B}^\dagger \otimes \mathbf{A})$vec$(\mathbf{X})$, where $\mathbf{X} \in \mathbb{R}^{Q \times P'}$.

## 2 The Matrix-Variate Gaussian Process

The matrix-variate Gaussian process (MV-GP) is a collection of random variables defined by their joint distribution for finite index sets. Let $\mathsf{M} \ni m$ be the set representing the rows (tasks) and $\mathsf{N} \ni n$ be the set representing the columns (examples), with sizes $|\mathsf{M}| = M$ and $|\mathsf{N}| = N$. Let $X \sim \mathcal{GP}(\phi, \mathcal{K})$, where $\mathcal{GP}(\phi, \mathcal{K})$ denotes the MV-GP with mean function $\phi$ and covariance function $\mathcal{K}$. As with the scalar GP, the MV-GP is completely specified by its mean function and its covariance function. These are defined as:

$$\phi(m,n) = \mathbb{E}[X(m,n)]$$
$$\mathcal{K}((m,n),(m',n')) = \mathbb{E}[(X(m,n) - \phi(m,n))(X(m',n') - \phi(m',n'))],$$

where $\mathbb{E}[\cdot]$ is the expected value. For a finite index set $\mathsf{M} \times \mathsf{N}$, define the matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$ such that $x_{m,n} = X(m,n)$, then vec$(\mathbf{X})$ is a distributed as a multivariate Gaussian with mean vec$(\mathbf{\Phi})$ and covariance matrix $\mathbf{K}$, i.e., vec$(\mathbf{X}) \sim \mathcal{N}(\text{vec}(\mathbf{\Phi}), \mathbf{K})$, where $\phi_{m,n} = \phi(m,n)$, $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$, and $k_{(m,n),(m',n')} = \mathcal{K}((m,n)(m',n'))$, $\mathbf{K} \in \mathbb{R}^{MN \times MN}$.

The covariance function of the prior MV-GP is assumed to have Kronecker product structure [7], [11]. The Kronecker product prior covariance captures the assumption that the prior covariance between matrix entries can be decomposed as the product of the row and column covariances. The Kronecker prior covariance assumption is a useful restriction as: (i) it improves computational tractability, enabling the model to scale to larger problems than may be possible with a full joint prior covariance, (ii) the regularity imposed by the separability assumption improves the reliability of parameter estimates even with significant data sparsity, e.g., when the observed data consists of a single matrix (sub-)sample, and (iii) row-wise and column-wise prior covariance functions are often the only prior information

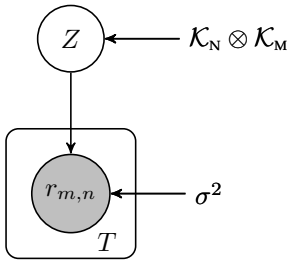


Fig. 1. Plate model of the matrix-variate Gaussian process. $Z(m,n)$ is the hidden noise-free matrix entry.

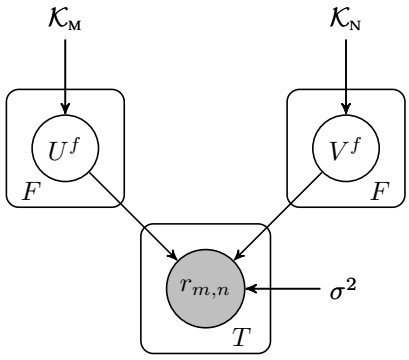


Fig. 2. Plate model of the factor Gaussian process.

available. A closely related concept is kernel MTL with *separable kernels*. This is a special case of vector valued regularized function estimation where the joint kernel decomposes into the product of the row kernel and the column kernel [6], [7], [25]. Learning in these models is analogous to inference for the MV-GP, with the prior row (resp. column) covariance matrix used as row (resp. column) kernels.

Define the row covariance (kernel) function $\mathcal{K}_\mathsf{M} : \mathsf{M} \times \mathsf{M} \mapsto \mathbb{R}$ and the column covariance function $\mathcal{K}_\mathsf{N} : \mathsf{N} \times \mathsf{N} \mapsto \mathbb{R}$. The joint covariance function of the MV-GP with Kronecker covariance decomposes into product form as $\mathcal{K}((m,n),(m',n')) = \mathcal{K}_\mathsf{M}(m,m')\mathcal{K}_\mathsf{N}(n,n')$, or equivalently, $\mathcal{K} = \mathcal{K}_\mathsf{N} \otimes \mathcal{K}_\mathsf{M}$. Hence, for the random variable $X \sim \mathcal{GP}(\phi, \mathcal{K}_\mathsf{N} \otimes \mathcal{K}_\mathsf{M})$ and a finite index set $\mathsf{M} \times \mathsf{N}$, $\mathrm{vec}(\mathbf{X}) \sim \mathcal{N}(\mathrm{vec}(\mathbf{\Phi}), \mathbf{K}_\mathsf{N} \otimes \mathbf{K}_\mathsf{M})$, where $\mathbf{K}_\mathsf{M} \in \mathbb{R}^{M \times M}$ is the row covariance matrix and and $\mathbf{K}_\mathsf{N} \in \mathbb{R}^{N \times N}$ is the column covariance matrix.

This definition also extends to finite subsets that are not complete matrices. Given any finite subset $\mathsf{T} = \{t = (m,n) \,|\, m \in \mathsf{M}, n \in \mathsf{N}\}$, where $T = |\mathsf{T}| \leq M \times N$, the vector $\mathbf{x} = [x_{t_1} \dots x_{t_T}]$ is distributed as $\mathbf{x} \sim \mathcal{N}(\mathbf{\Phi}_\mathsf{T}, \mathbf{K})$. The vector $\mathbf{\Phi}_\mathsf{T} = [\phi(1) \dots \phi(T)] \in \mathbb{R}^T$ are arranged from the entries of the mean matrix corresponding to the set $t \in \mathsf{T}$, and $\mathbf{K}$ is the covariance matrix evaluated only on pairs $t, t' \in \mathsf{T} \times \mathsf{T}$.

Our goal is to estimate an unknown response matrix $\mathbf{R} \in \mathbb{R}^{M \times N}$ with $M$ rows and $N$ columns. We assume observed data consisting of a subset of the matrix entries $\boldsymbol{r} = [r_{t_1} \dots r_{t_T}]$ collected into a vector. Note that $\mathsf{T} \subset \mathsf{M} \times \mathsf{N}$; hence, the data represents a partially observed matrix. Our generative assumption proceeds as follows (see Fig. 1):

1) Draw the function $Z$ from a zero mean MV-GP $Z \sim \mathcal{GP}(0, \mathcal{K}_\mathsf{N} \otimes \mathcal{K}_\mathsf{M})$.
2) Given $z_{m,n} = Z(m,n)$, draw each observed response independently: $r_{m,n} \sim \mathcal{N}(z_{m,n}, \sigma^2)$.

Hence, $\mathbf{Z} \in \mathbb{R}^{M \times N}$ with entries $z_{m,n} = Z(m,n)$ may be interpreted as the latent noise-free matrix. The inference task is to estimate the posterior distribution $Z|\mathcal{D}$, where $\mathcal{D} = \{\boldsymbol{r}, \mathsf{T}\}$. The posterior distribution is again a Gaussian process, i.e., $Z|\mathcal{D} \sim \mathcal{GP}(\phi, \Sigma)$, with mean and covariance functions:

$$\phi(m,n) = \mathbf{K}_\mathsf{T}(m,n)[\mathbf{K} + \sigma^2 \mathbf{I}_T]^{-1} \boldsymbol{r} \tag{1}$$

$$\Sigma((m,n),(m',n')) = \tag{2}$$
$$k((m,n),(m',n')) - \mathbf{K}_\mathsf{T}(m,n)[\mathbf{K} + \sigma^2 \mathbf{I}_T]^{-1} \mathbf{K}_\mathsf{T}(m,n)^\dagger.$$

where $\boldsymbol{r} \in \mathbb{R}^T$ corresponds to the vector of responses for all training data indexes $(m,n) \in \mathsf{T}$. The covariance function $\mathbf{K}_\mathsf{T}(m,n)$ corresponds to the sampled covariance matrix between the index $(m,n)$ and all training data indexes $(m',n') \in \mathsf{T}$, $\mathbf{K}$ is the covariance matrix between all pairs $(m,n),(m',n') \in \mathsf{T} \times \mathsf{T}$, and $\mathbf{I}_T$ is the $T \times T$ identity matrix. The closed form follows directly from the definition of a MV-GP as a scalar GP [9] with appropriately vectorized variables. The model complexity scales with the number of observed samples $T$. Storing the kernel matrix requires $\mathcal{O}(T^2)$ memory, and the naïve inference implementation requires $\mathcal{O}(T^3)$ computation.

Although the matrix-variate Gaussian process approach results in closed form inference, it does not model in low rank matrix structure. The factor GP is a hierarchical Gaussian process model that attempts to address this deficiency in the MV-GP [23], [24]. With a fixed model rank $F$, the generative model for the factor GP is as follows (see Fig. 2):

1) For each $f \in \{1 \dots F\}$, draw row functions: $U^f \sim \mathcal{GP}(0, \mathcal{K}_\mathsf{M})$. Let $\boldsymbol{u}_m \in \mathbb{R}^F$ with entries $u_m^f = U^f(m)$.
2) For each $f \in \{1 \dots F\}$, draw column functions: $V^f \sim \mathcal{GP}(0, \mathcal{K}_\mathsf{N})$. Let $\boldsymbol{v}_n \in \mathbb{R}^F$ with $v_n^f = V^f(n)$.
3) Draw each matrix entry independently: $r_{m,n} \sim \mathcal{N}(\boldsymbol{u}_m^\dagger \boldsymbol{v}_n, \sigma^2) \;\forall (m,n) \in \mathsf{T}$.

where $\boldsymbol{u}_m$ is the $m^{\text{th}}$ row of $\mathbf{U} = [\boldsymbol{u}^1 \dots \boldsymbol{u}^F] \in \mathbb{R}^{M \times F}$, and $\boldsymbol{v}_n$ is the $n^{\text{th}}$ row of $\mathbf{V} = [\boldsymbol{v}^1 \dots \boldsymbol{v}^F] \in \mathbb{R}^{N \times F}$. The maximum-a-posteriori (MAP) estimates of $\mathbf{U}$ and $\mathbf{V}$ can be computed as the solution of the following optimization problem:

$$\mathbf{U}^*, \mathbf{V}^* = \underset{\mathbf{U},\mathbf{V}}{\arg\min} \frac{1}{\sigma^2} \sum_{(m,n) \in T} (r_{m,n} - \boldsymbol{u}_m^\dagger \boldsymbol{v}_n)^2 + \mathrm{tr}(\mathbf{U}^\dagger \mathbf{K}_\mathsf{M}^{-1} \mathbf{U}) + \mathrm{tr}(\mathbf{V}^\dagger \mathbf{K}_\mathsf{N}^{-1} \mathbf{V}) \tag{3}$$

where $\mathrm{tr}(\mathbf{X})$ is the trace of the matrix $\mathbf{X}$. However, the joint posterior distribution of $\{\mathbf{U}, \mathbf{V}\}$ and the distribution of $\mathbf{Z} = \mathbf{U}\mathbf{V}^\dagger$ are not Gaussian, and the required expectations and posterior distributions are quite challenging to characterize. A Laplace approximation by proposed by [23] and [24] utilized sampling techniques. Statistically, the factor GP may be interpreted as the sum of rank-one factor matrices. Hence, as the rank $F \to \infty$, the law of large numbers can be used to show that the distribution of $Z$ converges to $\mathcal{GP}\left(0, \mathcal{K}_{\mathsf{N}} \otimes \mathcal{K}_{\mathsf{M}}\right)$ [23].

### 2.1 Spectral norms of compact operators

The mean function of the MV-GP is an element of the Hilbert space defined by the kernels (covariances). We provide a brief overview of some relevant background required for defining this representation and for defining relevant spectral norms of compact operators. We will focus on the MV-GP with Kronecker prior covariance. Our exposition is closely related to the approach outlined in [27]. Further details may be found in [28].

Let $\mathcal{H}_{\mathcal{K}_{\mathsf{M}}}$ denote the Hilbert space of functions induced by the row kernel $\mathcal{K}_{\mathsf{M}}$. Similarly, let $\mathcal{H}_{\mathcal{K}_{\mathsf{N}}}$ denote the Hilbert space of functions induced by the column kernel $\mathcal{K}_{\mathsf{N}}$. Let $\boldsymbol{x} \in \mathcal{H}_{\mathcal{K}_{\mathsf{M}}}$ and $\boldsymbol{y} \in \mathcal{H}_{\mathcal{K}_{\mathsf{N}}}$ define (possibly infinite dimensional) feature vectors. The mean function the MV-GP is defined by a linear map $W : \mathcal{H}_{\mathcal{K}_{\mathsf{M}}} \mapsto \mathcal{H}_{\mathcal{K}_{\mathsf{N}}}$, i.e., the bilinear form on $\mathcal{H}_{\mathcal{K}} = \mathcal{H}_{\mathcal{K}_{\mathsf{M}}} \times \mathcal{H}_{\mathcal{K}_{\mathsf{N}}}$ given by:

$$\phi(m, n) = \langle \boldsymbol{x}_m, W \boldsymbol{y}_n \rangle_{\mathcal{H}_{\mathcal{K}_{\mathsf{M}}}}$$

Let $\mathcal{B}$ denote the set of compact bilinear operators mapping $\mathcal{H}_{\mathcal{K}_{\mathsf{M}}} \mapsto \mathcal{H}_{\mathcal{K}_{\mathsf{N}}}$. A compact linear operator $W \in \mathcal{B}$ admits a spectral decomposition [27] with singular values given by $\{\xi_i(W)\}$.

**The trace norm** is given by the *ell*-1 norm on the spectrum of $W$:

$$\|\phi\|_{\mathcal{H}_{\mathcal{K}}, *} = \sum_{i=1}^{D} \xi_i(W) \tag{4}$$

When the dimensions are finite, (4) is the trace norm of the matrix $\boldsymbol{W} \in \mathbb{R}^{D_{\mathsf{M}} \times D_{\mathsf{N}}}$. This norm has been widely applied to several machine learning tasks including multitask learning [29], [30] and recommender systems [31]. In addition to the trace norm, a common regularizer is the induced **Hilbert norm** given by the *ell*-2 norm on the spectrum of $W$:

$$\|\phi\|^2_{\mathcal{H}_{\mathcal{K}}} = \sum_{i=1}^{D} \xi_i^2(W) \tag{5}$$

(5) is equivalent to the matrix Frobenius norm for finite dimensional $\mathbf{W} \in \mathbb{R}^{D_{\mathsf{M}} \times D_{\mathsf{N}}}$ computed as:

$$\|\mathbf{W}\|^2_{\mathrm{F}} = \sum_{i=1}^{\min(D_{\mathsf{M}}, D_{\mathsf{N}})} \xi_i^2(\mathbf{W})$$

Let $L(\phi, \boldsymbol{r}, \mathsf{T})$ represent the loss function for a finite set of training data points $\mathsf{T} \in \mathsf{M} \times \mathsf{N}$ and $Q(\phi)$ be a spectral regularizer. We define the regularized risk functional:

$$L(\phi, \boldsymbol{r}, \mathsf{T}) + \lambda Q(\phi)$$

where $\lambda \geq 0$ is the regularization constant. A representer theorem exists, i.e., the function $\phi$ that optimizes the regularized risk can be represented as a finite weighted sum of the kernel functions evaluated on training data [27]. Employing this representer theorem, the optimizing function can be computed as:

$$\begin{aligned} \phi(m, n) &= \sum_{m' \in \mathsf{M}} \sum_{n' \in \mathsf{N}} \alpha_{m', n'} \, \mathcal{K}_{\mathsf{M}}(m, m') \mathcal{K}_{\mathsf{N}}(n, n') \\ &= \mathbf{K}_{\mathsf{M}}(m) \boldsymbol{A} \mathbf{K}_{\mathsf{M}}(n)^\dagger \end{aligned} \tag{6}$$

where $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ is a parameter matrix, $\mathbf{K}_{\mathsf{M}}(m)$ is the kernel matrix evaluated between $m$ and $m' \in \mathsf{M}$, i.e., the $m^{\text{th}}$ row of $\mathbf{K}_{\mathsf{M}}$, and $\mathbf{K}_{\mathsf{N}}(n)$ is the kernel matrix evaluated between $n$ and all $n' \in \mathsf{N}$.

**Computing the norms:** The Hilbert norm can be computed as:

$$\begin{aligned} \|\phi\|^2_{\mathcal{H}_{\mathcal{K}}} &= \mathrm{vec}(\boldsymbol{A})^\dagger (\mathbf{K}_{\mathsf{N}} \otimes \mathbf{K}_{\mathsf{M}}) \mathrm{vec}(\boldsymbol{A}) \\ &= \mathrm{tr}(\boldsymbol{A}^\dagger \mathbf{K}_{\mathsf{M}} \boldsymbol{A} \mathbf{K}_{\mathsf{N}}). \end{aligned} \tag{7}$$

The trace norm can be computed using a basis transformation approach [27] or by using the low rank "variational" approximation [27], [30].

**Basis transformation:** With the index set fixed, define bases $\mathbf{G}_{\mathsf{M}} \in \mathbb{R}^{M \times D_{\mathsf{M}}}$ and $\mathbf{G}_{\mathsf{N}} \in \mathbb{R}^{N \times D_{\mathsf{N}}}$ such that $\mathbf{K}_{\mathsf{M}} = \mathbf{G}_{\mathsf{M}} \mathbf{G}_{\mathsf{M}}^\dagger$ and $\mathbf{K}_{\mathsf{N}} = \mathbf{G}_{\mathsf{N}} \mathbf{G}_{\mathsf{N}}^\dagger$. One such basis is the square root of the kernel matrix $\mathbf{G}_{\mathsf{M}} = \mathbf{K}_{\mathsf{M}}^{\frac{1}{2}}$ and $\mathbf{G}_{\mathsf{N}} = \mathbf{K}_{\mathsf{N}}^{\frac{1}{2}}$. When the feature space is finite dimensional, the feature matrices $\mathbf{X}_{\mathsf{M}} \in \mathbb{R}^{M \times D_{\mathsf{M}}}$ and $\mathbf{X}_{\mathsf{N}} \in \mathbb{R}^{N \times D_{\mathsf{N}}}$ are also an appropriate basis. The mean function can be reparametrized as $\phi(m, n) = \mathbf{G}_{\mathsf{M}}(m) \mathbf{B} \mathbf{G}_{\mathsf{N}}(n)^\dagger$, where $\mathbf{B} \in \mathbb{R}^{D_{\mathsf{M}} \times D_{\mathsf{N}}}$. Now, the trace norm is given directly by the trace norm of the parameter matrix, i.e., $\|\phi\|_{\mathcal{H}_{\mathcal{K}}, *} = \|\mathbf{B}\|_*$.

**Low rank "variational" approximation:** The trace norm can also be computed using the low rank approximation. This is sometimes known as the variational approximation of the trace norm [30].

$$\|\phi\|_{\mathcal{H}_{\mathcal{K}}, *} = \underset{\phi = \langle u, v \rangle}{\arg\min} \frac{1}{2} \sum_{f=1}^{F} \left( \left\|u^f\right\|^2_{\mathcal{H}_{\mathcal{K}_{\mathsf{M}}}} + \left\|v^f\right\|^2_{\mathcal{H}_{\mathcal{K}_{\mathsf{N}}}} \right) \tag{8}$$

where $\langle u, v \rangle = \sum_{f=1}^{F} u^f v^f$. This approach is exact when $F$ is larger than the true rank of $\phi$. Note that this is the same regularization that is required for MAP inference with the factor GP model (3). Hence, when $F$ is sufficiently large, the regularizer in the factor GP model is the trace norm. Unfortunately, it is difficult to select an appropriate rank a-priori, and no such claims exist when $F$ is insufficiently large. With finite dimensions, the variational approximation of the trace norm is given by:

$$\|\mathbf{W}\|_* = \underset{\mathbf{W} = \mathbf{U}\mathbf{V}^\dagger}{\arg\min} \frac{1}{2} \left( \|\mathbf{U}\|^2_{\mathrm{F}} + \|\mathbf{V}\|^2_{\mathrm{F}} \right)$$

where $\mathbf{U} \in \mathbb{R}^{D_{\mathsf{M}} \times F}$ and $\mathbf{V} \in \mathbb{R}^{D_{\mathsf{N}} \times F}$. This sum of factor norms has proven effective for the regularization of matrix factorization models and other latent factor problems [32].

## 3 Trace norm constrained inference for the MV-GP

A generative model for low rank matrices has proven to be a challenging problem. We are unaware of any (non-hierarchical) distributions in the literature that generate sample matrices of low rank. Hierarchical models have been proposed, but such models introduce issues such as non-convexity and non-identifiability of parameter estimates. Instead of seeking a generative model for low rank matrices, we propose a variational inference approach. We constrain the inference of the MV-GP such that expected value of the approximate posterior distribution has a constrained trace norm (and is generally of low rank). In contrast to standard variational inference, this constraint is enforced in order to extract structure.

The goal of inference is to estimate of the posterior distribution $p(Z|\mathcal{D})$. We propose approximate inference using the log likelihood lower bound [33]:

$$\ln p(\mathbf{y}|\mathcal{D}) \geq \mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{y}, \mathbf{Z})] - \mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{Z})] \tag{9}$$

Our approach is to restrict the search to the space of Gaussian processes $q(Z) = \mathcal{GP}(\psi, S)$ subject to a trace norm constraint $\|\psi\|_{\mathcal{K},*} \leq C$ as defined in (4). With no loss of generality, we assume a set of rows $\mathsf{M}$ and columns $\mathsf{N}$ of interest so $\mathsf{T} \in \mathsf{M} \times \mathsf{N}$. Let $\mathbf{Z} \in \mathbb{R}^{M \times N}$ be the matrix of hidden variables.

Given finite indexes, $q$ is a Gaussian distribution $q(\mathbf{z}) = \mathcal{N}(\boldsymbol{\psi}, \mathbf{S})$ where $\mathbf{z} = \mathrm{vec}(\mathbf{Z}) \in \mathbb{R}^{M \times N}$, $\boldsymbol{\phi} = \mathrm{vec}(\boldsymbol{\Psi}) \in \mathbb{R}^{M \times N}$, and $\mathbf{S} \in \mathbb{R}^{MN \times MN}$. Let $\mathbf{P} \in \mathbb{R}^{T \times MN}$ be a permutation matrix such that $\mathbf{S}_T = \mathbf{P}\mathbf{S}\mathbf{P}^\dagger$ is the covariance matrix of the subset of observed entries $t \in \mathsf{T}$, and $\mathbf{K}_T = \mathbf{P}\mathbf{K}\mathbf{P}^\dagger$ is the prior covariance of the corresponding subset of entries. Evaluating expectations, the lower bound (9) results in the following inference cost function (omitting terms independent of $\boldsymbol{\psi}$ and $\mathbf{S}$):

$$\begin{aligned}\max_{\boldsymbol{\psi}, \mathbf{S}} \; & -\frac{1}{2\sigma^2} \sum_{m,n \in \mathsf{T}} (r_{m,n} - \psi_{m,n})^2 - \frac{1}{2\sigma^2}\mathrm{tr}(\mathbf{S}_T) \\ & -\frac{1}{2}\boldsymbol{\psi}^\dagger \mathbf{K}^{-1}\boldsymbol{\psi} - \frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{S}) + \ln|\mathbf{S}| \\ & \text{s.t. } \|\psi\|_{\mathcal{K},*} \leq C\end{aligned}$$

where $|\mathbf{X}|$ is the determinant of matrix $\mathbf{X}$.

First, we compute gradients with respect to $\mathbf{S}$. After setting the gradients to zero, we compute:

$$\begin{aligned}\mathbf{S}^* &= \left(\mathbf{K}^{-1} + \frac{1}{\sigma^2}\mathbf{P}^\dagger\mathbf{P}\right)^{-1} \\ &= \mathbf{K} - \mathbf{K}\mathbf{P}^\dagger\left(\mathbf{K}_T + \frac{1}{\sigma^2}\mathbf{I}_T\right)^{-1}\mathbf{P}\mathbf{K},\end{aligned} \tag{10}$$

The second equality is a consequence of the matrix inversion lemma. Interestingly, (10) is identical the posterior covariance of the unconstrained MV-GP (2).

Next, we collect the terms involving the mean. This results in the optimization problem:

$$\begin{aligned}\boldsymbol{\psi}^* = \arg\min_{\boldsymbol{\psi}} \; & \frac{1}{2\sigma^2} \sum_{m,n \in \mathsf{T}} (r_{m,n} - \psi_{m,n})^2 + \frac{1}{2}\boldsymbol{\psi}^\dagger \mathbf{K}^{-1}\boldsymbol{\psi} \\ & \text{s.t. } \|\psi\|_{\mathcal{H}_\mathcal{K},*} \leq C\end{aligned} \tag{11}$$

This is a convex regularized least squares problem with a convex constraint set. Hence, (11) is convex, and $\boldsymbol{\psi}^*$ is unique. Using the Kronecker identity, we can rewrite the cost function in parameter matrix form. We can also replace the trace constraint with the equivalent regularizer weighed by $\mu$. Multiplying through by $\sigma^2$ leads to the equivalent cost:

$$\begin{aligned}\boldsymbol{\Psi}^* = \arg\min_{\Psi} \; & \frac{1}{2} \sum_{m,n \in \mathsf{T}} (r_{m,n} - \psi_{m,n})^2 \\ & + \frac{\sigma^2}{2}\mathrm{tr}(\boldsymbol{\Psi}^\dagger \mathbf{K}_{\mathsf{M}}^{-1} \boldsymbol{\Psi} \mathbf{K}_{\mathsf{N}}^{-1}) + \mu\sigma^2 \|\psi\|_{\mathcal{H}_\mathcal{K},*},\end{aligned} \tag{12}$$

Applying the representation (6), we recover the parametric form of the mean function $\psi \in \mathcal{K}_{\mathsf{N}} \otimes \mathcal{K}_{\mathsf{M}}$ as $\boldsymbol{\Psi} = \mathbf{K}_{\mathsf{M}}\boldsymbol{A}\mathbf{K}_{\mathsf{N}}$ where $\boldsymbol{A} \in \mathbb{R}^{M \times N}$. We may also solve for $\boldsymbol{A}$ directly:

$$\begin{aligned}\boldsymbol{A}^* = \arg\min_{\boldsymbol{A}} \; & \frac{1}{2} \sum_{m,n \in \mathsf{T}} (r_{m,n} - (\mathbf{K}_{\mathsf{M}}\boldsymbol{A}\mathbf{K}_{\mathsf{N}})_{m,n})^2 \\ & + \frac{\sigma^2}{2}\mathrm{tr}(\boldsymbol{A}^\dagger \mathbf{K}_{\mathsf{M}} \boldsymbol{A} \mathbf{K}_{\mathsf{N}}) + \mu\sigma^2 \|\psi(\boldsymbol{A})\|_{\mathcal{H}_\mathcal{K},*}.\end{aligned} \tag{13}$$

where $\psi(\boldsymbol{A})$ is the mean function corresponding to the parameter $\boldsymbol{A}$ (see (6)). The representation of the mean function in functional form is useful for avoiding repeated optimization when testing a trained model with different evaluation sets.

The approximate posterior distribution is itself a finite index set representation of an underlying Gaussian process.

**Theorem 1.** *The posterior distribution $q = \mathcal{N}(\boldsymbol{\psi}, \mathbf{S})$ is the finite index set representation of the Gaussian process $\mathcal{GP}(\psi, S)$ where the mean function $\psi$ is given by* (13) *and the covariance function $S$ is given by* (2)*. $g = \mathcal{GP}(\psi, S)$ is the unique posterior distribution that maximizes the lower bound of the log likelihood* (9) *subject to the trace constraint $\|\psi\|_{\mathcal{K},*} \leq C$.*

**Sketch of proof:** Uniqueness of the solution follows from (9), which is jointly convex in $\{\boldsymbol{\psi}, \mathbf{S}\}$. To show that the posterior distribution is a Gaussian process, we simply need to show that for a fixed training set $\mathcal{D}$, the posterior distribution of the superset $(\mathsf{M} \times \mathsf{N}) \cup (m', n')$ has the same mean function and covariance function. These follow directly from the solution of (13) and from (2) (see (10)). In addition to showing uniqueness, Theorem 1 shows how the trained model can be extended to evaluate the posterior distribution of data points not in training.

In the case where a basis for $\mathbf{K}_{\mathsf{M}}$ and $\mathbf{K}_{\mathsf{N}}$ can be found, (11) may be solved using the matrix trace norm approach

directly (see section 2.1):

$$\mathbf{B}^* = \arg\min_{\mathbf{B}} \frac{1}{2} \sum_{m,n \in \mathsf{T}} \left(r_{m,n} - (\mathbf{G}_{\mathrm{M}} \mathbf{B} \mathbf{G}_{\mathrm{N}}^{\dagger})_{m,n}\right)^2 + \frac{\sigma^2}{2} \|\mathbf{B}\|_{\mathrm{F}}^2 + \mu\sigma^2 \|\mathbf{B}\|_* . \quad (14)$$

where $\mathbf{G}_{\mathrm{M}} \in \mathbb{R}^{M \times D_{\mathrm{M}}}$ is the basis for $\mathbf{K}_{\mathrm{M}}$ and $\mathbf{G}_{\mathrm{N}} \in \mathbb{R}^{N \times D_{\mathrm{N}}}$ is the basis for $\mathbf{K}_{\mathrm{N}}$. $\mathbf{B} \in \mathbb{R}^{D_{\mathrm{M}} \times D_{\mathrm{N}}}$ is the estimated parameter matrix. The mean function is then given by $\psi_{m,n} = (\mathbf{G}_{\mathrm{M}} \mathbf{B} \mathbf{G}_{\mathrm{N}}^{\dagger})_{m,n}$.

**Spectral elastic net regularization:** The regularization that results from the constrained inference has an interesting interpretation as the spectral elastic net norm. As discussed in section 2.1, the mean function may be represented as $\psi(m,n) = \langle \boldsymbol{x}_m, W \boldsymbol{y}_n \rangle_{\mathcal{H}_{\mathcal{K}_{\mathrm{M}}}}$ for $\boldsymbol{x} \in \mathcal{H}_{\mathcal{K}_{\mathrm{M}}}$ and $\boldsymbol{y} \in \mathcal{H}_{\mathcal{K}_{\mathrm{N}}}$. The spectral elastic net is given as a weighed sum of the *ell*-2 norm and the *ell*-1 norms on the spectrum $\{\xi_i(W)\}$:

$$Q_{\alpha,\beta}(\psi) = \alpha \sum_{i=1}^{D} \xi_i^2(W) + \beta \sum_{i=1}^{D} \xi_i(W) \quad (15)$$

where $\alpha$ and $\beta \geq 0$ are weighting constants. The naming is intentionally suggestive of the analogy to the elastic net regularizer [26]. The elastic net regularizer is a weighted sum of the *ell*-2 norm and the *ell*-1 norms of the parameter vector in a linear model. The elastic net is a tradeoff between smoothness, encouraged by the *ell*-2 norm, and sparsity, encouraged by the *ell*-1 norm. The elastic net is particularly useful when learning with correlated features. The spectral elastic net has similar properties. The Hilbert norm encourages smoothness over the spectrum, while the trace norm encourages spectral sparsity, i.e., low rank. To the best of our knowledge, this combination of norms is novel, both in the matrix estimation literature and in the kernelized MTL literature. When the dimensions are finite, (15) is given by a weighted sum of the trace norm and the Frobenius norm of the parameter matrix.

We propose a parametrization of the mean function inference inspired by the elastic net [26]. Let $\lambda = \sigma^2(1+\mu)$ and $\alpha = \frac{\mu\sigma^2}{\sigma^2(1+\mu)}$ where $\lambda \geq 0$ and $\alpha \in [0,1]$. The loss function (14) can be parametrized as:

$$\mathbf{B}^* = \arg\min_{\mathbf{B}} \frac{1}{2} \sum_{m,n \in \mathsf{T}} \left(r_{m,n} - (\mathbf{G}_{\mathrm{M}} \mathbf{B} \mathbf{G}_{\mathrm{N}}^{\dagger})_{m,n}\right)^2 + \frac{\lambda(1-\alpha)}{2} \|\mathbf{B}\|_{\mathrm{F}}^2 + \lambda\alpha \|\mathbf{B}\|_* . \quad (16)$$

The same parametrization can also be applied to the equivalent representations given in (12) and (13). This spectral elastic net parametrization clarifies the tradeoff between the trace norm and the Hilbert norm. The trace norm is recovered when $\alpha = 1$, and the Hilbert norm is recovered for $\alpha = 0$. The spectral elastic net approach is also useful for speeding up the computation with warm-start i.e. for a fixed $\alpha$, we may employ warm-start for decreasing values of $\lambda$. Computation of the spectral elastic net norm follows directly from the Hilbert and trace norms. From the variational approximation of the trace norm (8), it is clear that MAP inference for the factor GP (3) is equivalent to inference for the mean of the trace constrained MV-GP (12) in the special case where $\alpha = 1$ (assuming that the non-convex optimization (3) achieves the global maximum).

**Non-zero mean prior:** To simplify the explanation, we have assumed so far that the prior Gaussian process has a zero mean. The non-zero mean case is a straightforward extension [9]. We include a short discussion for completeness. Let $b_{m,n}$ represent the mean parameter of the Gaussian process prior, i.e., $Z \sim \mathcal{GP}(b, \mathcal{K}_{\mathrm{N}} \otimes \mathcal{K}_{\mathrm{M}})$. The posterior covariance estimate remains the same, and the posterior mean computation requires the same optimization, but with the observation $r_{m,n}$ replaced by $\tilde{r}_{m,n} = r_{m,n} - b_{m,n}$. The resulting posterior mean must then be shifted by the bias, i.e., $\mathbb{E}[Z_{m,n}|\mathcal{D}] = \psi_{m,n} + b_{m,n}$. If desired, this parameter may be easily estimated. Suppose we choose to model a row-wise bias. Let $\mathsf{T}_m = \{(m,n) | (m',n) \in T,\, m' = m\}$, then solving the straightforward optimization, we find that the the row bias estimate is given by:

$$b_m = \frac{1}{|\mathsf{T}_m|} \sum_{n | m,n \in T_m} r_{m,n} - \psi_{m,n}.$$

## 4 Bipartite Ranking

Bipartite ranking is the task of learning an ordering for items drawn from two sets, known as the positive set and the negative set, such that the items in the positive set are ranked ahead of the items in the negative set [1], [2], [3], [4]. Many models for bipartite ranking attempt to optimize the pair-wise mis-classification cost, i.e., the model is penalized for each pair of data points where the positive labeled item is ranked lower than the negative labeled item. Although this approach has proven effective, the required computation is quadratic in the number of items. This quadratic computation cost limits the applicability of pair-wise bipartite ranking to large scale problems.

More recently, researchers have shown that it may be sufficient to optimize a classification loss, such as the exponential loss or the logistic loss, directly to solve the bipartite ranking problem [3], [4]. This is also known as the point-wise approach in the ranking literature. In contrast to the point-wise and pair-wise approach, we propose a *list-wise* bipartite ranking model. The list-wise approach learns a ranking model for the entire set of items and has gained prominence in the learning to rank literature [34], [35] as it comes with strong theoretical guarantees and has been shown to have superior empirical performance.

Our approach is inspired by monotone retargeting (MR) [35], a recent method for adapting regression to ranking tasks. Although many ranking models are trained to predict the relevance scores, there is no need

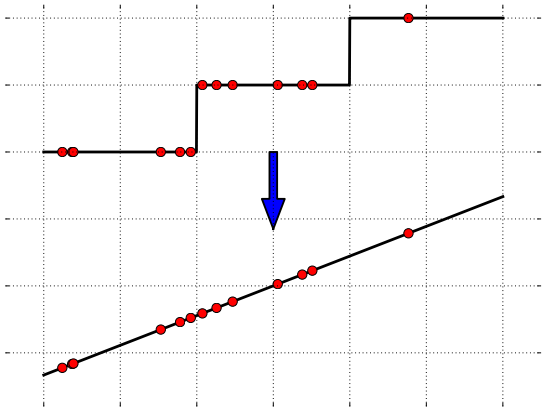

Fig. 3. Monotone re-targeting searches for an order preserving transformation of the target scores that may be easier for the regressor to fit.

to fit scores exactly. Any scores that induce the correct ordering will suffice. MR jointly optimizes over monotonic transformations of the target scores and an underlying regression function (see Fig. 3). We will show that maximum likelihood parameter estimation in the proposed model is equivalent to learning the target scores in MR. Though we show this equivalence for the special case of bipartite ranking with square loss, the relation holds for more general Bregman divergences and ranking tasks. This extension is beyond the scope of this paper. In addition to improving performance, MR has favorable statistical and optimization theoretic properties, particularly when combined with a Bregman divergence such as squared loss. To the best of our knowledge, ours is the first generative model for list-wise bipartite ranking.

### 4.1 Background

Let $\mathbb{B} = \{+1, -1\}$, and let $\mathbb{B}^d_\downarrow$ be the set of binary isotonic vectors (binary vectors in sorted order), i.e., any $\boldsymbol{v} \in \mathbb{B}^d_\downarrow$ satisfies $\boldsymbol{v} \in \mathbb{B}^d$ and $v_i \geq v_j \ \forall j > i$. Similarly, let $\mathbb{R}^d_\downarrow$ be the set of real valued isotonic vectors, i.e., any $\boldsymbol{v} \in \mathbb{R}^d_\downarrow$ satisfies $\boldsymbol{v} \in \mathbb{R}^d$ and $v_i \geq v_j \ \forall j > i$, then $\boldsymbol{v}$ satisfies *partial* order. We state that $\boldsymbol{v}$ satisfies *total* order or *strict isotonicity* when the ordering is a strict inequality, i.e., $v_i > v_j \ \forall j > i$. We denote a vector in sorted order as $\vec{\boldsymbol{v}} = \text{sort}(\boldsymbol{v})$.

Compatibility is a useful concept for capturing the match between the sorted order of two vectors.

**Definition 2** (Compatibility [34]). *$\boldsymbol{u}$ is compatible with the sorted order of $\vec{\boldsymbol{v}}$ (denoted as $\boldsymbol{u} \rightsquigarrow \vec{\boldsymbol{v}}$) if for every pair of indexes $(i, j)$, $\vec{\boldsymbol{v}}_i \geq \vec{\boldsymbol{v}}_j$ implies $\boldsymbol{u}_i \geq \boldsymbol{u}_j$.*

Compatibility is an asymmetric relationship, i.e., $\boldsymbol{u} \rightsquigarrow \vec{\boldsymbol{v}} \not\Rightarrow \vec{\boldsymbol{v}} \rightsquigarrow \boldsymbol{u}$. It follows that sorted vectors always satisfy compatibility, i.e., if $\vec{\boldsymbol{u}}$ and $\vec{\boldsymbol{v}}$ are two sorted vectors, then by definition 2, $\vec{\boldsymbol{u}} \rightsquigarrow \vec{\boldsymbol{v}}$. Compatibility is straightforward to check when the target vector is binary. Let $\boldsymbol{u} \in \mathbb{R}^d$ and $\vec{\boldsymbol{v}} \in \mathbb{B}^d_\downarrow$, and let $k$ be the be the number of $+1$'s in the sorted vector $\vec{\boldsymbol{v}}$, i.e., the threshold for transition between $+1$ and $-1$. Then $\boldsymbol{u} \rightsquigarrow \vec{\boldsymbol{v}}$ implies that:

$$\exists\, b \in \mathbb{R} \quad \text{s.t.} \quad \min_{1 \leq i \leq k} u_i \geq b \geq \max_{k < j \leq d} u_j. \tag{17}$$

There are several ways to permute a sorted binary vector $\vec{\boldsymbol{y}} \in \mathbb{B}^d_\downarrow$ while keeping all its values the same. These are permutations that separately re-order the $+1$s at the top of $\vec{\boldsymbol{y}}_m$ and the $-1$s at the bottom. Given $\vec{\boldsymbol{y}} = \text{sort}(\boldsymbol{y})$, we represent the set of permutations that do not change the value the sorted $\vec{\boldsymbol{y}}$ as $\Gamma = \{\gamma(\cdot) \mid \gamma(\vec{\boldsymbol{y}}) = \vec{\boldsymbol{y}}\}$. It follows that the set $\Gamma$ contains all permutations that satisfy $\gamma(\vec{\boldsymbol{v}}) \rightsquigarrow \vec{\boldsymbol{y}}$. In other words, all $\boldsymbol{v}$ that satisfy $\boldsymbol{v} \rightsquigarrow \vec{\boldsymbol{y}}$ can be represented as $\boldsymbol{v} = \gamma(\vec{\boldsymbol{v}})$ for some $\gamma \in \Gamma$.

We propose a representation for compatible vectors that reduces to permutations of isotonic vectors.

**Proposition 3.** *Let $\vec{\boldsymbol{v}} \in \mathbb{B}^d_\downarrow$. Any $\boldsymbol{u} \in \mathbb{R}^d$ that satisfies $\boldsymbol{u} \rightsquigarrow \vec{\boldsymbol{v}}$ can be represented by $\boldsymbol{u} = \gamma(\vec{\boldsymbol{u}})$ where $\gamma \in \Gamma$, the set $\Gamma = \{\gamma(\cdot) \mid \gamma(\vec{\boldsymbol{v}}) = \vec{\boldsymbol{v}}\}$ and $\vec{\boldsymbol{u}} \in \mathbb{R}^d_\downarrow$.*

**Sketch of proof:** First, we note that by definition of compatibility for binary vectors (17), any permutation $\gamma \in \Gamma$ satisfies $\gamma(\vec{\boldsymbol{u}}) \rightsquigarrow \vec{\boldsymbol{v}}$. Next, we note that the sorted order is a member of the permutation set. This representation is unique when $\vec{\boldsymbol{u}}$ satisfies strict ordering.

The set $\mathbb{R}^d_\downarrow$ is a convex cone. To see this, note that the convex composition $\boldsymbol{x} = \alpha\boldsymbol{u} + (1-\alpha)\boldsymbol{v}$, $\alpha \in [0, 1]$ of two isotonic vectors $\boldsymbol{u} \in \mathbb{R}^d_\downarrow$ and $\boldsymbol{v} \in \mathbb{R}^d_\downarrow$ preserves isotonicity. Further, any scaling $\alpha\boldsymbol{x}$ where $\alpha > 0$ preserves the ordering. Let $\Delta^d$ be the set of probability distributions, i.e., $\forall\, \boldsymbol{x} \in \Delta^d$, $x_i \geq 0$ and $\sum_{i=1}^d x_i = 1$. The set of probability distributions in sorted order is given by $\Delta^d_\downarrow = \mathbb{R}^d_\downarrow \cap \Delta^d$ so for each $\boldsymbol{x} \in \Delta^d_\downarrow$, $\boldsymbol{x} \in \Delta^d$ and $x_i \geq x_j \, \forall i > j$.

**Lemma 4** (Representation of $\Delta^d_\downarrow$ [35])**.** *The set $\Delta^d_\downarrow$ of all discrete probability distributions of dimension $d$ that are in descending order is the image $\mathbf{C}\boldsymbol{x}$ where $\boldsymbol{x} \in \Delta^d$ and $\mathbf{C}$ is an upper triangular matrix generated from the vector $\boldsymbol{v} = \{1, \frac{1}{2} \cdots \frac{1}{T}\}$ such that $\mathbf{C}(i,:) = \{0\}^{i-1} \times \boldsymbol{v}(i:)$*

### 4.2 Generative model

Let $y_{m,n} \in \mathbb{B}$ be the label for item $n$ in $m^{\text{th}}$ task and let $\mathsf{T}_m = \{n \,|\, (m,n) \in \mathsf{T}\}$ be the set of items in $m^{\text{th}}$ task so $|\mathsf{T}_m| = T_m$. We define the negative set as $\mathsf{D}^- = \{(m,n) \in \mathsf{T} \,|\, y_{m,n} = -1\}$ and the positive set as $\mathsf{D}^+ = \{(m,n) \in \mathsf{T} \,|\, y_{m,n} = +1\}$. For the $m^{\text{th}}$ task, the negative set is defined as $\mathsf{D}^-_m = \{n | (m,n) \in \mathsf{D}^-\}$ and the positive set as $\mathsf{D}^+_m = \{n | (m,n) \in \mathsf{D}^+\}$ so that $\mathsf{T}_m = \mathsf{D}^+_m \cup \mathsf{D}^-_m$. The vector of labels for the $m^{\text{th}}$ task are given by $\boldsymbol{y}_m \in \mathbb{B}^{T_m}$.

We propose the following generative model for $\boldsymbol{y}_m$:

$$p(\boldsymbol{y}_m | \boldsymbol{r}_m) \propto \prod_{l \in \mathsf{D}^+_m} \prod_{l' \in \mathsf{D}^-_m} \mathbb{1}_{[r_{m,l} \geq r_{m,l'}]}. \tag{18}$$

where $\mathbb{1}_{[\cdot]}$ is the indicator function defined as:

$$\mathbb{1}_{[b]} = \begin{cases} 1 & \text{if } b \text{ evaluates to true}\,, \\ 0 & \text{otherwise.} \end{cases}$$

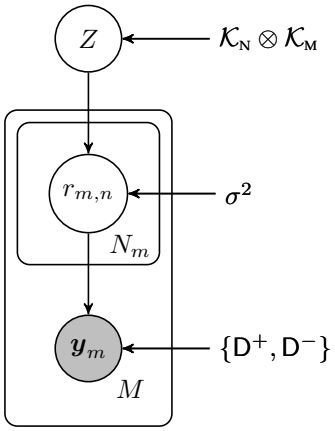


Fig. 4. Plate model of generative bipartite ranking with the latent matrix-variate Gaussian process.

For clarity, we have suppressed the dependence of $p(\boldsymbol{y}_m|\boldsymbol{r}_m)$ on the sets $\{\mathsf{D}_m^+, \mathsf{D}_m^-\}$.

It is instructive to compare the form of the generative model (18) to the area under the ROC curve (AUC) given by the fraction of correctly ordered pairs:

$$\frac{1}{\left|\mathsf{D}_m^+\right|\left|\mathsf{D}_m^-\right|} \sum_{l \in \mathsf{D}_m^+} \sum_{l' \in \mathsf{D}_m^-} \left( \mathbb{1}_{[r_{m,l} > r_{m,l'}]} + \frac{1}{2} \mathbb{1}_{[r_{m,l} = r_{m,l'}]} \right). \tag{19}$$

We note that $p(\boldsymbol{y}_m|\boldsymbol{r}_m)$ is nonzero if and only if $\boldsymbol{r}_m$ satisfies the ordering defined by $\{\mathsf{D}_m^+, \mathsf{D}_m^-\}$. It follows that any vector $\boldsymbol{r}_m$ s.t. $p(\boldsymbol{y}_m|\boldsymbol{r}_m)$ is nonzero also maximizes the AUC.

We can now combine the bipartite ranking model with the latent regression model. The full generative model proceeds as follows (see Fig. 4):

1) Draw the latent variable $Z$ from a zero mean MV-GP as $Z \sim \mathcal{GP}\left(0, \mathcal{K}_{\mathsf{N}} \otimes \mathcal{K}_{\mathsf{M}}\right)$.
2) Given $z_{m,n} = Z(m,n)$, draw each score vector independently as $r_{m,n} \sim \mathcal{N}\left(z_{m,n}, \sigma^2\right)$.
3) For each task $m \in \mathsf{M}$, draw the observed response vector $p(\boldsymbol{y}_m|\boldsymbol{r}_m)$ as given by (18).

### 4.3 Inference and parameter estimation

We utilize variational inference to train the underlying multitask regression model and maximum likelihood to estimate the parameters of the bipartite ranking model. This is equivalent to the variational approximation $q(\boldsymbol{r}, Z) = \mathbb{1}_{[\boldsymbol{r}=\boldsymbol{r}^*]} q(Z)$ where $\boldsymbol{r} = \{\boldsymbol{r}_m\}$. The variational lower bound of the log likelihood (9) is given by:

$$\ln p(\boldsymbol{y}|\mathcal{D}) \geq \ln p(\boldsymbol{y}|\boldsymbol{r}) + \mathbb{E}_{\mathbf{Z}}[\ln p(\boldsymbol{r}, \mathbf{Z})] - \mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{Z})]$$

where $\boldsymbol{y} = \{\boldsymbol{y}_m\}$. As outlined in section 3, we restrict our search to the space for $q(Z)$ of Gaussian processes $q(Z) = \mathcal{GP}(\psi, S)$ subject to a trace norm constraint $\|\psi\|_{\mathcal{K},*} \leq C$. Evaluating expectations (and ignoring constant terms independent of $\{\boldsymbol{r}, \psi, S\}$) results in the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{r}, \boldsymbol{\psi}, \mathbf{S}} & - \sum_{m \in \mathsf{M}} \sum_{l \in \mathsf{D}_m^+} \sum_{l' \in \mathsf{D}_m^-} \ln \left( \mathbb{1}_{[r_{m,l} \geq r_{m,l'}]} \right) \\ & + \frac{1}{2\sigma^2} \sum_{m,n \in \mathsf{T}} (r_{m,n} - \psi_{m,n})^2 + \frac{1}{2\sigma^2} \mathrm{tr}(\mathbf{S}_T) \\ & + \frac{1}{2} \boldsymbol{\psi}^\dagger \mathbf{K}^{-1} \boldsymbol{\psi} + \frac{1}{2} \mathrm{tr}(\mathbf{K}^{-1} \mathbf{S}) - \ln |\mathbf{S}| \\ & \text{s.t. } \|\psi\|_{\mathcal{K},*} \leq C \end{aligned} \tag{20}$$

Inference and parameter estimation follow an alternating optimization scheme. We alternately optimize each of the parameters $\{\boldsymbol{r}, \psi, S\}$ till a local optima is reached. Following section 3, it is straightforward to show that the optimal $S$ is given in closed form (10) and is independent of $\{\boldsymbol{r}, \psi\}$. Hence, model training requires alternating between optimizing $\boldsymbol{r}^*|\psi$ and optimizing $\psi^*|\boldsymbol{r}$. We will show that (20) is convex, and the alternating optimization approach achieves the global optimum. The optimization for $\psi^*|\boldsymbol{r}$ follows directly from the discussion in section 3. Hence, we will focus our efforts on the optimization of $\boldsymbol{r}^*|\psi$.

Collecting the terms of (20) that are dependent on $\boldsymbol{r}$ results in the following loss function for $\boldsymbol{r}|\psi$:

$$\min_{\boldsymbol{r}} \overbrace{- \sum_{m \in \mathsf{M}} \sum_{l \in \mathsf{D}_m^+} \sum_{l' \in \mathsf{D}_m^-} \ln \left( \mathbb{1}_{[r_{m,l} \geq r_{m,l'}]} \right)}^{\text{Order violation penalty}} + \underbrace{\frac{1}{2\sigma^2} \sum_{m,n \in \mathsf{T}} (r_{m,n} - \psi_{m,n})^2}_{\text{Square loss}}$$

The first term in the loss penalizes violations of order. In fact, the first term evaluates to infinity if any of the binary order constraints are violated. Hence, to maximize the log likelihood, the variables $\boldsymbol{r}$ must satisfy the order constraints $\{\boldsymbol{r}_m \rightsquigarrow \vec{\boldsymbol{y}}_m \ \forall m \in \mathsf{M}\}$. This interpretation suggests a constrained optimization approach:

$$\min_{\{\boldsymbol{r}_m | \boldsymbol{r}_m \rightsquigarrow \vec{\boldsymbol{y}}_m\}} \frac{1}{2} \sum_{l \in \mathsf{T}_m} (r_{m,l} - \psi_{m,l})^2 \quad \forall m \in \mathsf{M} \tag{21}$$

Note that this loss decomposes task-wise. Hence, the proposed approach results in a list-wise ranking model. We also note that the independence between tasks means that the optimization is embarrassingly parallel.

The constrained score vectors $\{\boldsymbol{r}_m | \boldsymbol{r}_m \rightsquigarrow \vec{\boldsymbol{y}}_m\}$ can be optimized efficiently using the inner representation outlined in Proposition 3. One issue that arises is that the cost function (20) is not invariant to scale. Hence, the loss can be reduced just by scaling its arguments down. To avoid this degeneracy, we must constrain the score vectors away from $\mathbf{0}$. We achieve this by constraining the score vectors to the ordered simplex $\Delta_{\downarrow}^{T_m}$, as it is a convex set and satisfies the requirement $\mathbf{0} \notin \Delta_{\downarrow}^{T_m}$. Applying Lemma 4, the score is given by $\boldsymbol{r}_m = \gamma(\vec{\boldsymbol{r}}_m) = \gamma(\mathbf{C}\boldsymbol{x}_m)$ for $\boldsymbol{x}_m \in \Delta^{T_m}$.

**initialize** $\psi$, $\{\boldsymbol{x}_m\}$, $\{\gamma_m\}$
**repeat**
  Update $\psi^*|\boldsymbol{r}$ by solving (12).
  **for all** $m \in \mathsf{M}$ **do**
    Update $\gamma_m^*|\boldsymbol{\psi}_m$ by block sorting (Lemma 5).
    Update $\boldsymbol{x}_m^*|\gamma_m, \boldsymbol{\psi}_m$ by solving (23).
  **end for**
**until** converged
**return** $\psi$, $\{\boldsymbol{x}_m\}$, $\{\gamma_m\}$
Compute $S$ using (10)

Algorithm 1: Variational inference and maximum likelihood parameter estimation.

Let $\boldsymbol{\psi}_m \in \mathbb{R}^{T_m}$ be the score vector ordered to satisfy $\boldsymbol{\psi}_m = [\{\psi(m,l)\,\forall\, l \in \mathsf{D}_m^+\}|\{\psi(m,l')\,\forall\, l' \in \mathsf{D}_m^-\}]$. The ordering of the score vector is not unique. The loss function can now be written as:

$$\min_{\boldsymbol{x}\in\Delta^{T_m}} \min_{\pi\in\Gamma_m} \frac{1}{2}\left\|\pi(\mathbf{C}\boldsymbol{x}) - \boldsymbol{\psi}_m\right\|_2^2 \quad \forall\, m \in \mathsf{M} \tag{22}$$

This is exactly equivalent to:

$$\min_{\boldsymbol{x}\in\Delta^{T_m}} \min_{\gamma\in\Gamma_m} \frac{1}{2}\left\|\mathbf{C}\boldsymbol{x} - \gamma(\boldsymbol{\psi}_m)\right\|_2^2 \quad \forall\, m \in \mathsf{M} \tag{23}$$

The equivalence can be shown simply by setting $\gamma(\cdot) = \pi^{-1}(\cdot)$. We present both forms as it provides some flexibility when implementing the algorithm. We optimize (23) by alternating optimization. We first optimize the vector $\boldsymbol{x}_m^*$ and then optimize the permutation order $\gamma_m^*$. The overall optimization combining the variational inference and maximum likelihood parameter estimation is presented in Algorithm 1. The probability vector $\boldsymbol{x}_m$ can be optimized efficiently using the exponentiated gradient (EG) algorithm [36] or other simplex-constrained least squares solvers and can be embarrassingly parallelized over the tasks $\mathsf{T}_m$. Optimization of $\gamma_m$ requires optimizing over all permutations of the vector. This may be naïvely solved by expensive enumeration or by solving a combinatorial assignment problem.

**Lemma 5** (Optimality of sorting [35])**.** *If $x_1 \geq x_2$ and $y_1 \geq y_2$, then $\left\|\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix} - \begin{bmatrix}\boldsymbol{y}_1\\ \boldsymbol{y}_2\end{bmatrix}\right\|_2^2 \leq \left\|\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix} - \begin{bmatrix}\boldsymbol{y}_2\\ \boldsymbol{y}_1\end{bmatrix}\right\|_2^2$ and $\left\|\begin{bmatrix}\boldsymbol{y}_1\\ \boldsymbol{y}_2\end{bmatrix} - \begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}\right\|_2^2 \leq \left\|\begin{bmatrix}\boldsymbol{y}_2\\ \boldsymbol{y}_1\end{bmatrix} - \begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix}\right\|_2^2$.*

Lemma 5 implies that selecting the pair-wise sorted order for equivalent items minimizes the loss. Lemma 5 can then be extended to $\boldsymbol{y} \in \mathbb{R}^d$ using induction over $d$. Hence, selecting $\gamma_m$ as the sorted ordering in each block of equivalent values $\{\mathsf{D}_m^+, \mathsf{D}_m^-\}$ minimizes the loss.

The set of compatible vectors as defined by Proposition 3 is a convex cone, i.e., the convex composition $\boldsymbol{z} = \alpha\boldsymbol{u} + (1-\alpha)\boldsymbol{v}$, $\alpha \in [0,1]$ of vectors in the set remains in the set, and any scaling $\alpha\boldsymbol{z}$ where $\alpha > 0$ preserves compatibility. To show convexity of parameter estimation, it remains to show that the combination of optimizing $\boldsymbol{x}_m$ and optimizing $\gamma_m$ minimizes (21). This is shown using the following lemma.

**Lemma 6** (Convexity of parameter estimation [35])**.** *Let $\boldsymbol{r}$ be partitioned into two sets such that $\boldsymbol{r}_1 = \{r_k,\ \forall k \in \mathsf{D}^+\}$ and $\boldsymbol{r}_2 = \{r_k,\ \forall k \in \mathsf{D}^-\}$, and let:*

$$\begin{bmatrix}\boldsymbol{r}_1\\ \boldsymbol{r}_2\end{bmatrix}^* = \underset{\substack{\boldsymbol{r}_i'\in\Pi(\boldsymbol{r}_i)\\ \boldsymbol{r}_1'\geq\boldsymbol{r}_2'}}{\arg\min} \left\|\begin{bmatrix}\boldsymbol{r}_1'\\ \boldsymbol{r}_2'\end{bmatrix} - \begin{bmatrix}\boldsymbol{z}_1\\ \boldsymbol{z}_2\end{bmatrix}\right\|_2^2,$$

*where $\Pi(\boldsymbol{r}_i)$ is the set of all permutations of the vector $\boldsymbol{r}_i$ and $\boldsymbol{r}_1' \geq \boldsymbol{r}_2'$ represents element wise inequality, then $\boldsymbol{r}_i^*$ is isotonic with $\boldsymbol{z}_i$ $\forall i = 1, 2$.*

We can now consider the global properties of the variational inference and maximum likelihood parameter estimation.

**Lemma 7** (Joint convexity)**.** *The variational inference and parameter estimation given by* (20) *is jointly convex in $\{\psi, S, \boldsymbol{r}\}$. Alternating optimization (Algorithm 1) recovers the global optimum.*

**Sketch of proof:** Recall that squared loss is jointly convex in both of its arguments. In addition, the components $\{\psi, S\}$ and $\boldsymbol{r}$ are in separate convex sets. Hence to show global convexity, it is sufficient to show that (20) is convex separately in $\{\psi, S\}$ (by Theorem 1) and $\boldsymbol{r}$ (by Lemma 6). It follows from the joint convexity of (20) that the alternating optimization of Algorithm 1 recovers the global optimum.

The proposed model is trained to estimate bipartite ranking scores for each task and the underlying multitask latent regression distribution. Item rankings are predicted by sorting the expected noise-free scores of the trained model $\mathbb{E}[z_{m,n}|\mathcal{D}] = \psi(m,n)$.

## 5 Experiments

This section details the experiments comparing the performance of the proposed model applied to the disease-gene prioritization task. We evaluated the modeling performance on association data curated from the OMIM database [37] by the authors of [17] and data we curated ourselves. We partitioned each dataset into five-fold cross validation sets. The model was trained on $4$ of the $5$ sets and tested on the held out set. The results presented are the averaged 5-fold cross validation performance. Great care was taken to train all the models on the same datasets. Hence the results represent performance differences due to either the low rank modeling, the list-wise bipartite ranking model, or both.

**Baseline (ProDiGe [17]):** We compared our proposed model to ProDiGe which, to the best of our knowledge, is the state of the art in the disease-gene prioritization literature. ProDiGe estimates the prioritization function using multitask support vector machines trained with gene kernel and disease kernel information. Parameter selection for ProDiGe was performed as suggested by the authors [17].

**OMIM dataset:** The OMIM dataset [37] is a curated database of known human disease-gene associations ($4178$ associations in the provided dataset). We derived

the gene-gene interaction graph using data from HumanNet [38]. We selected all genes with one or more connections in the network and all diseases with one or more genetic associations. This resulted in a disease-gene matrix with $M = 3,210$ diseases, $N = 13,614$ genes and $T = 3,636$ known associations (data sparsity .0083%). In addition, the gene-gene graph contained $433,224$ known gene-gene links. We note the extreme sparsity of this matrix, and the resulting difficulty of the ranking task. Such sparse datasets are typical in the disease-gene domain. The OMIM dataset did not contain a disease graph; hence, we were unable to test the generalization of the methods to new diseases.

**Curated dataset:** We curated a large disease-gene association dataset. The set of genes were defined using the NCBI ENTREZ Gene database [39], and the set of diseases were defined using the "Disease" branch of the NIH Medical Subject Heading (MeSH) ontology [40]. We extracted co-citations of these genes and diseases in the PubMed/Medline database [41] to identify positive instances of disease-gene associations. We derived our gene-gene interaction graph using data from HumanNet [38] and our disease-disease similarity graph from the MeSH ontology. This resulted in a set of $250,190$ observed interactions, $21,243$ genes and $4,496$ diseases. We selected all genes with one or more connections in the gene-gene graph and all diseases with with one or more connections in the disease-disease graph. This resulted in a dataset with $M = 4,495$ diseases, $N = 13,614$ genes and $T = 224,091$ known associations (data sparsity $0.36\%$). The resulting disease network contained $13,922$ links, and the gene network contained $433,224$ links.

We were unable to run ProDiGe on the full dataset due to insufficient memory for storing the kernel matrix. Instead, we trained ProDiGe and the MV-GP models on a randomly selected 5% subsample of the associations. We also provide results for the MV-GP models trained on the full dataset. We performed two kinds of experiments for the curated dataset. The first experiment (**known diseases**) tests the ranking ability of the model for associations selected randomly over the matrix. The second experiment (**new diseases**) tests the generalization ability of the model for new diseases not observed in the training set. For the known disease experiments, the cross validation associations were randomly selected over the matrix. For the new disease experiments, the cross validation was performed row-wise, i.e., we selected training set diseases and test set diseases.

**Model Setup:** The proposed model was trained using the alternating optimization approach (Algorithm 1). The trace constrained mean function was estimated using the cost function (16). The model was trained using our implementation of the algorithm outlined in [42]. Like other large scale trace constrained matrix optimizers, [42] maintains a low rank representation. The rank is estimated automatically by the optimizer. We found that employing a row bias improved the model performance, so we learned row biases while training. Note that the row offsets do not change the ranking and hence are not required for testing.

We selected the hyperparameter $\lambda = s * \lambda_{\max}$ with 30 values of $s$ logarithmically spaced between $10^{-3}$ and 1.0. Let $F(\mathbf{B})$ be the loss function. Then $\lambda_{\max}$ is the maximum singular value of $\frac{dF(\mathbf{B})}{d\mathbf{B}}\big|_{\mathbf{B}=\mathbf{0}}$. The optimization returns the zero matrix for any $\lambda > \lambda_{\max}$. [30]. We used warm start to speed up the computation for decreasing values of $s$. We selected $\alpha \in \{1,\ 0.8,\ 0.6,\ 0.4,\ 0\}$.

We implemented the full rank Gaussian process model ($\alpha = 0$) by keeping the kernels as separate row and column kernels. This allowed us to scale the model to the larger datasets at the expense of more computations. We observed that the full rank model required a significant amount of computation time. This observation provides further motivation for the low rank approach. The full rank Gaussian process was trained directly using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm.

**Sampling unknown negative items:** Recall that the observed data only consists of known associations. Following ProDiGe [17] we sampled "negative" observations randomly over the disease-gene association matrix. We sampled 10 different negatively labeled item sets. All models are trained with the positive set combined with one of the negative labeled sets. The model scores are computed by averaging the scores over the 10 trained models. All algorithms were trained using the same samples.

**Covariance/Kernels:** The covariances for the MV-GP prior and the kernels for ProDiGe were computed from gene graph $\mathcal{G}_{\text{M}}$ and the disease graphs $\mathcal{G}_{\text{N}}$. We performed preliminary experiments with a large class of graph kernels [43] and selected the exponential kernel. We briefly outline kernel generation for the gene kernel. Let $\mathbf{A}_{\text{M}}$ be the adjacency matrix for the gene-gene graph. We computed the normalized Laplacian matrix as $\mathbf{L}_{\text{M}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}_{\text{M}}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{I}$ is the identity matrix and $\mathbf{D}$ is a diagonal matrix with entries $\mathbf{D}_{i,i} = (\mathbf{A}_{\text{M}}\mathbf{1})_i$. The exponential kernel is given by $\mathbf{K}'_{\text{M}} = \exp(-\mathbf{L}_{\text{M}})$. Following the suggestion in [17] and preliminary experiments, we observed an improvement in performance with the identity matrix added to the exponential kernel. Hence the final kernel is given by $\mathbf{K}_{\text{M}} = \exp(-\mathbf{L}_{\text{M}}) + \mathbf{I}$. The disease kernel generation was obtained using the same approach when a disease-disease graph was available. All algorithms were trained using the same kernel matrices.

**Metrics:** Experimental validation of disease-gene associations in a laboratory can be time consuming and costly, so only a small set of the top ranked predictions are of practical interest. Hence, we focus on metrics that capture the ranking behavior of the model at the top of the ranked list. In addition, all metrics are computed on the test set after removing all relevant genes that had been observed in the training set removed. All metrics are computed per disease and then averaged over all the diseases in the test set. Let $\vec{\boldsymbol{g}}_l$ denote the labels of item (gene) $l$ as sorted by the predicted scores of the trained

regression model, and let $G_m = |\mathrm{D}_m^+| = \sum_l \mathbb{1}_{[\vec{g}_l=1]}$ be the total number of relevant genes for disease $m$ in the test data after removing relevant genes observed in the training data. The metrics computed are as follows:

1) Area under the ROC curve (AUC) (19). This measures the overall ranking performance of the model.
2) The precision at $k \in \{1, 2, \ldots, 100\}$ computes the fraction of relevant genes retrieved out off all retrieved genes at position $k$.

$$\mathrm{P}_{@k} = \frac{\sum_{l=1}^{k} \mathbb{1}_{[\vec{g}_l=1]}}{k}.$$

3) The recall at $k \in \{1, 2, \ldots, 100\}$ computes the fraction of relevant genes retrieved out off all relevant genes.

$$\mathrm{R}_{@k} = \frac{\sum_{l=1}^{k} \mathbb{1}_{[\vec{g}_l=1]}}{\min(G_m, k)}.$$

4) Mean average precision at $k = 100$ (MAP100) computed as the mean of the average precision at $k = 100$. The average precision is given as:

$$\mathrm{AP}_{@k} = \frac{\sum_{l=1}^{k} \mathbb{1}_{[\vec{g}_l=1]} \mathrm{P}_{@l}}{\min(G_m, k)}$$

The $\mathrm{MAP}_{@100}$ metric was used for model selection over the cross validation runs. To reduce notation, MAP refers to $\mathrm{MAP}_{@100}$ in all results. Higher values reflect better performance for the AUC, $\mathrm{P}_{@k}$, $\mathrm{R}_{@k}$ and MAP metrics, and their maximum value is 1.0.

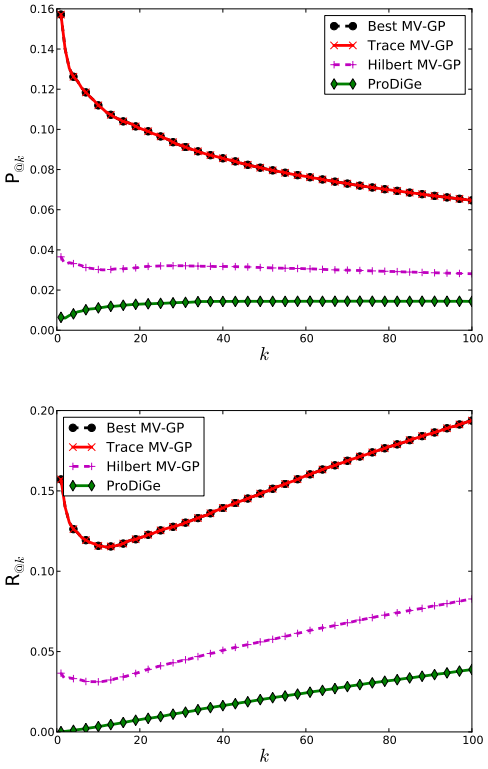


Fig. 5. Curated data (known diseases, 5% subsample) experiment results: precision (top) and recall (bottom) curves $@k = 1, 2, \ldots, 100$. Best and Trace curves overlap.

### 5.1 Discussion

We present performance results for ProDiGe, the standard MV-GP (*Hilbert*, $\alpha = 0$), the trace norm regularized MV-GP (*Trace*, $\alpha = 1$), and the best overall MV-GP model (*Best*).

Our first experiment was on the known disease prediction with the 5% subset of the curated data. This task is very challenging as the training data consisted of an average of less than 3 known associations out the possible $13,614$ per disease. The difficulty of this task is reflected in the performance results shown in Table 1 and Fig. 5. We found that the trace model had the same performance as the best MV-GP model, suggesting that the trace norm is an effective regularization in this case. We found that the trace regularization resulted in a significant improvement in performance across metrics compared to ProDiGe and the Hilbert models. We also experimented with predicting the gene ranking of new diseases not seen during training and found similar performance as shown in Table 2 and Fig. 6. As this is new disease prediction, none of the known genes on are removed from the test diseases. Interestingly, we found that this seems to improve the model performance as compared to the in-matrix prediction.

Next, we experimented with prediction on the full curated dataset predicting known diseases. We were unable run this experiment with ProDiGe due to memory limitations. Hence, only results from the proposed model are shown. The results are as shown in Table 3 and Fig. 7. As expected, we observed a significant improvement in performance by using the entire dataset. Similar results were observed for the new disease prediction as shown in Table 4 and Fig. 8. In all models, we observed that the trace norm constrained approach out-performed the standard full rank MV-GP model. We especially note the performance improvement at the top of the list, as these are the most important to the domain.

Our final experiment was on the OMIM dataset. The results are as shown in Table 5 and Fig. 9. These results are especially interesting as we found that the best overall model outperformed the trace model, and significantly outperformed all other model in terms of ranking at the top of the list. This suggests that the spectral elastic net regularizer may be most useful with significant data sparsity. ProDiGe out-performed the Hilbert model in terms of recall, but Hilbert model had the best overall ranking performance as measured by AUC. We are investigating this observation further, but preliminary investigation suggests that the metrics are more sensitive to small changes in order when the data is very sparse. The sparsity also explains the significant drop in $\mathrm{P}_{@k}$ as k grows.

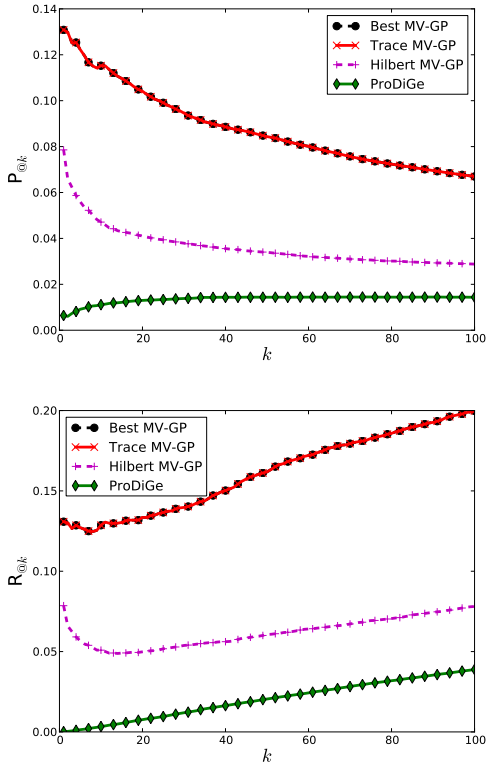


Fig. 6. Curated data (new diseases, 5% subsample) experiment results: precision (top) and recall (bottom) curves $@k = 1, 2, \ldots, 100$. Best and Trace curves overlap.

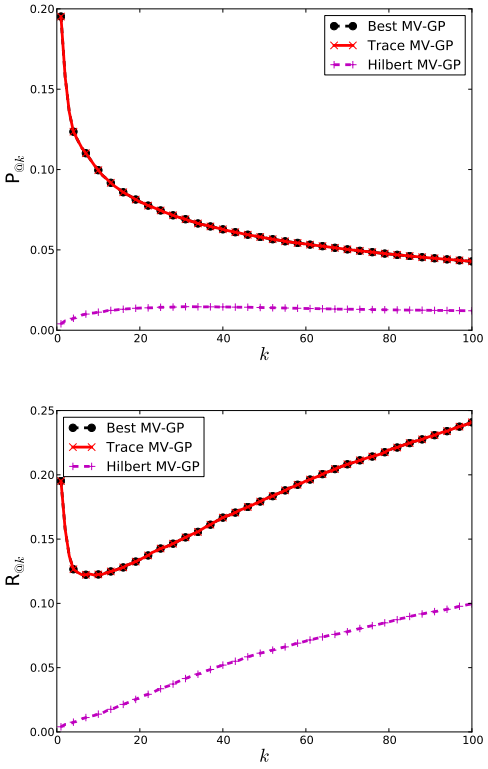


Fig. 7. Curated data (known diseases, full dataset) experiment results: precision (top) and recall (bottom) curves $@k = 1, 2, \ldots, 100$. Best and Trace curves overlap.

TABLE 1
Curated data experiment (known diseases, 5% subsample) avg. (std.) performance comparison.

| | Best | Trace | Hilbert | ProDiGe |
|---|---|---|---|---|
| AUC | **0.793 (0.002)** | **0.793 (0.002)** | 0.687 (0.002) | 0.716 (0.001) |
| MAP | **0.042 (0.003)** | **0.042 (0.003)** | 0.009 (0.001) | 0.003 (0.000) |
| $P_{@100}$ | **0.065 (0.001)** | **0.065 (0.001)** | 0.028 (0.001) | 0.014 (0.000) |
| $R_{@100}$ | **0.194 (0.001)** | **0.194 (0.001)** | 0.083 (0.003) | 0.039 (0.002) |

TABLE 2
Curated data experiment (new diseases, 5% subsample) avg. (std.) performance comparison.

| | Best | Trace | Hilbert | ProDiGe |
|---|---|---|---|---|
| AUC | **0.822 (0.014)** | **0.822 (0.014)** | 0.661 (0.018) | 0.716 (0.001) |
| MAP | **0.047 (0.009)** | **0.047 (0.009)** | 0.013 (0.004) | 0.003 (0.000) |
| $P_{@100}$ | **0.067 (0.014)** | **0.067 (0.014)** | 0.029 (0.009) | 0.014 (0.000) |
| $R_{@100}$ | **0.200 (0.019)** | **0.200 (0.019)** | 0.078 (0.011) | 0.039 (0.002) |

TABLE 3
Curated data experiment (known diseases, full dataset) avg. (std.) performance comparison.

| | Best | Trace | Hilbert |
|---|---|---|---|
| AUC | **0.869 (0.001)** | **0.869 (0.001)** | 0.782 (0.001) |
| MAP | **0.054 (0.001)** | **0.054 (0.001)** | 0.006 (0.000) |
| $P_{@100}$ | **0.043 (0.000)** | **0.043 (0.000)** | 0.012 (0.000) |
| $R_{@100}$ | **0.241 (0.001)** | **0.241 (0.001)** | 0.100 (0.001) |

TABLE 4
Curated data experiment (new diseases, full dataset) avg. (std.) performance comparison.

| | Best | Trace | Hilbert |
|---|---|---|---|
| AUC | **0.871 (0.009)** | **0.871 (0.009)** | 0.787 (0.015) |
| MAP | **0.080 (0.018)** | **0.080 (0.018)** | 0.013 (0.003) |
| $P_{@100}$ | **0.086 (0.021)** | **0.086 (0.021)** | 0.040 (0.010) |
| $R_{@100}$ | **0.255 (0.021)** | **0.255 (0.021)** | 0.125 (0.013) |

## 6 Conclusion

This paper proposes a novel hierarchical model for multitask bipartite ranking that combines a trace constrained matrix-variate Gaussian process and a bipartite ranking model. We showed that the trace constraint led to a mean function with low rank and discussed the spectral elastic net as the MAP regularizer that arises from this model.

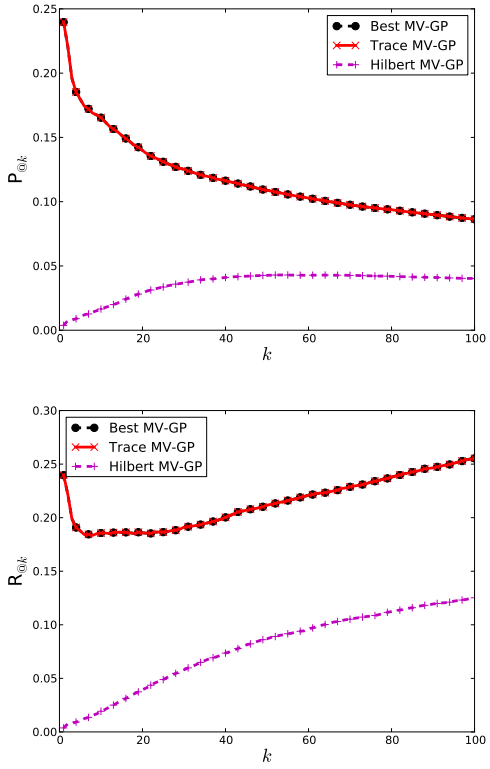


Fig. 8. Curated data (new diseases, full dataset) experiment results: precision (top) and recall (bottom) curves $@k = 1, 2, \ldots, 100$. Best and Trace curves overlap.

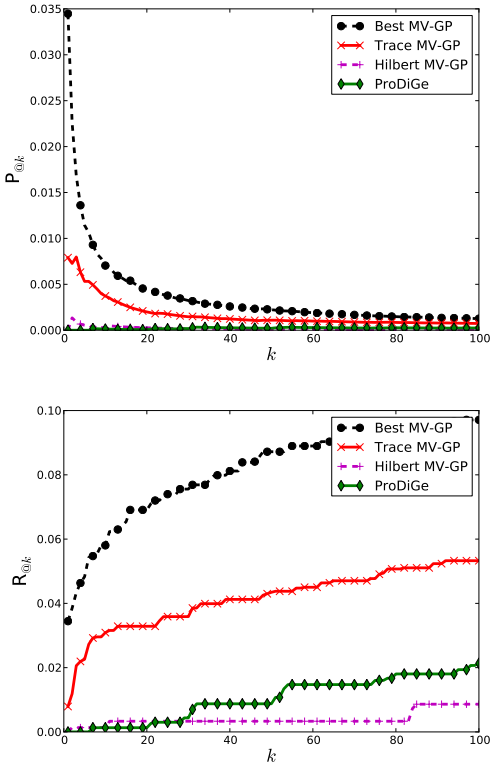


Fig. 9. OMIM data experiment results: precision (top) and recall (bottom) curves $@k = 1, 2, \ldots, 100$.

TABLE 5
OMIM data experiment avg. (std.) performance comparison.

| | Best | Trace | Hilbert | ProDiGe |
|---|---|---|---|---|
| AUC | 0.654 (0.028) | 0.649 (0.029) | **0.686 (0.016)** | 0.524 (0.018) |
| MAP | **0.041 (0.008)** | 0.015 (0.002) | 0.001 (0.001) | 0.001 (0.000) |
| $P_{@100}$ | **0.001 (0.000)** | 0.001 (0.000) | 0.000 (0.000) | 0.000 (0.000) |
| $R_{@100}$ | **0.097 (0.014)** | 0.053 (0.018) | 0.009 (0.003) | 0.021 (0.005) |

We showed that constrained variational inference for the Gaussian process combined with maximum likelihood parameter estimation for the ranking model was jointly convex. We applied the proposed model to the prioritization of disease-genes and found that the proposed model significantly improved performance over strong baseline models.

We plan to explore the trace norm constrained MV-GP and the spectral elastic net further and analyze their theoretical properties. We also plan to explore parameter estimation using the resulting constrained posterior distribution. In addition, we plan to investigate the applications of the constrained MV-GP to other tasks including multitask regression and collaborative filtering.

## Acknowledgments

Authors acknowledge support from NSF grant IIS 1016614. We thank Sreangsu Acharyya for helpful discussions on bipartite ranking. We also thank U. Martin Blom and Edward Marcotte for providing the OMIM data set.

## References

[1] C. Cortes and M. Mohri, "Auc optimization vs. error rate minimization," in *Advances in Neural Information Processing Systems*. MIT Press, 2004.

[2] S. Agarwal and P. Niyogi, "Stability and generalization of bipartite ranking algorithms," in *Proceedings of the Eighteenth Annual Conference on Computational Learning Theory (COLT)*. Springer, 2005, pp. 32–47.

[3] W. Kotlowski, K. Dembczynski, and E. Huellermeier, "Bipartite ranking through minimization of univariate loss," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ser. ICML '11, L. Getoor and T. Scheffer, Eds. New York, NY, USA: ACM, June 2011, pp. 1113–1120.

[4] W. Gao and Z.-H. Zhou, "On the consistency of auc optimization," *CoRR*, vol. abs/1208.0645, 2012.

[5] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2004, pp. 109–117.

[6] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *Journal of Machine Learning Research*, vol. 6, pp. 615–637, 2005.

[7] M. A. lvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: a review," *CoRR*, 2011.

[8] O. Stegle, C. Lippert, J. M. Mooij, N. D. Lawrence, and K. Borgwardt, "Efficient inference in matrix-variate gaussian models with iid observation noise," in *NIPS*, 2011, pp. 630–638.

[9] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning series)*. The MIT Press, Nov. 2005.

[10] K. Yu and W. Chu, "Gaussian process models for link analysis and transfer learning," in *NIPS*, 2008, pp. 1657–1664.

[11] K. Yu, J. Lafferty, S. Zhu, and Y. Gong, "Large-scale collaborative prediction using a nonparametric random effects model," in *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*. New York, NY, USA: ACM, 2009, pp. 1185–1192.

[12] E. Bonilla, K. M. Chai, and C. Williams, "Multi-task gaussian process prediction," in *NIPS 20*, 2008, pp. 153–160.

[13] NCBI. (1998) Genes and disease. Online. National Center for Biotechnology Information. Retrieved January 10, 2011. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK22183/

[14] M. I. McCarthy, G. R. Abecasis, L. R. Cardon, D. B. Goldstein, J. Little, J. P. Ioannidis, and J. N. Hirschhorn, "Genome-wide association studies for complex traits: consensus, uncertainty and challenges." *Nature reviews. Genetics*, vol. 9, no. 5, pp. 356–369, May 2008.

[15] O. Vanunu, O. Magger, E. Ruppin, T. Shlomi, and R. Sharan, "Associating genes and protein complexes with disease via network propagation," *PLoS Computational Biology*, vol. 6, no. 1, 2010.

[16] Y. Li and J. C. Patra, "Genome-wide inferring gene and phenotype relationship by walking on the heterogeneous network," *Bioinformatics*, vol. 26, pp. 1219–1224, May 2010.

[17] F. Mordelet and J.-P. Vert, "Prodige: Prioritization of disease genes with multitask machine learning from positive and unlabeled examples." *BMC Bioinformatics*, vol. 12, p. 389, 2011.

[18] N. Natarajan, U. M. Blom, A. Tewari, J. O. Woods, I. S. Dhillon, and E. M. Marcotte, "Predicting gene-disease associations using multiple species data," Department of Computer Science, University of Texas at Austin, Tech. Rep. TR-11-37, October 2011.

[19] C. Elkan and K. Noto, "Learning classifiers from only positive and unlabeled data," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '08. New York, NY, USA: ACM, 2008, pp. 213–220.

[20] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*, 2009, pp. 452–461.

[21] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *ICDM*, 2008, pp. 502–511.

[22] P. G. Sun, L. Gao, and S. Han, "Prediction of human disease-related gene clusters by clustering analysis," *International Journal of Biological Sciences*, vol. 7, no. 1, pp. 61–73, 2011.

[23] K. Yu, W. Chu, S. Yu, V. Tresp, and Z. Xu, "Stochastic relational models for discriminative link prediction," in *Advances in Neural Information Processing Systems 19*. Cambridge, MA: MIT Press, 2007, pp. 1553–1560.

[24] S. Zhu, K. Yu, and Y. Gong, "Stochastic relational models for large-scale dyadic data using mcmc," in *Advances in Neural Information Processing Systems 21*. Cambridge, MA: MIT Press, 2009, pp. 1993–2000.

[25] O. Koyejo and J. Ghosh, "A kernel-based approach to exploiting interaction-networks in heterogeneous information sources for improved recommender systems," in *HetRec '11*, 2011, pp. 9–16.

[26] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society, Series B*, vol. 67, pp. 301–320, 2005.

[27] J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert, "A new approach to collaborative filtering: Operator estimation with spectral regularization," *JMLR*, vol. 10, pp. 803–826, 2009.

[28] A. Berlinet and C. Thomas-Agnan, *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Boston, Dordrecht, London: Kluwer Academic Publishers, 2004.

[29] T. K. Pong, P. Tseng, S. Ji, and J. Ye, "Trace norm regularization: Reformulations, algorithms, and multi-task learning," *SIAM J. on Optimization*, vol. 20, no. 6, pp. 3465–3489, Dec. 2010.

[30] M. Dudík, Z. Harchaoui, and J. Malick, "Lifted coordinate descent for learning with trace-norm regularization." *AISTATS*, vol. 22, pp. 327–336, 2012.

[31] N. Srebro, J. D. M. Rennie, and T. S. Jaakola, "Maximum-margin matrix factorization," in *Advances in Neural Information Processing Systems 17*. MIT Press, 2005, pp. 1329–1336.

[32] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *NIPS*, 2008, pp. 1257–1264.

[33] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[34] P. Ravikumar, A. Tewari, and E. Yang, "On NDCG consistency of listwise ranking methods," in *Proceedings of 14th International Conference on Artificial Intelligence and Statistics*, ser. AISTATS, 2011.

[35] S. Acharyya, O. Koyejo, and J. Ghosh, "Learning to rank with bregman divergences and monotone retargeting," in *Proceedings of the 28th conference on Uncertainty in artificial intelligence*, ser. UAI '12, 2012.

[36] J. Kivinen and M. K. Warmuth, "Exponentiated gradient versus gradient descent for linear predictors," *Information and Computation*, vol. 132, 1995.

[37] V. A. McKusick, "Mendelian Inheritance in Man and its online version, OMIM." *American journal of human genetics*, vol. 80, no. 4, pp. 588–604, Apr. 2007.

[38] I. Lee, U. M. Blom, P. I. Wang, J. E. Shim, and E. M. Marcotte, "Prioritizing candidate disease genes by network-based boosting of genome-wide association data," *Genome Research*, vol. 21, no. 7, pp. 1109–1121, May 2011.

[39] D. R. Maglott, J. Ostell, K. D. Pruitt, and T. A. Tatusova, "Entrez gene: gene-centered information at NCBI," *Nucleic Acids Research*, vol. 39, no. Database-Issue, pp. 52–57, 2011.

[40] N. L. of Medicine, "Medical Subject Headings," http://www.nlm.nih.gov/mesh/.

[41] ——, "PubMed," http://www.ncbi.nlm.nih.gov/pubmed/.

[42] S. Laue, "A hybrid algorithm for convex semidefinite optimization," in *ICML*, 2012.

[43] A. J. Smola and I. Kondor, "Kernels and regularization on graphs," in *COLT*, 2003.



**Oluwasanmi Koyejo** is a PhD student working on data mining and machine learning at The University of Texas at Austin and advised by Dr. Joydeep Ghosh. He received his B.S. Degree in Electrical Engineering with a minor in Statistics from the New Jersey Institute of Technology (NJIT) and completed a M.S. in Electrical Engineering at the University of Texas at Austin with a focus on machine learning applied to wireless communications.

**Cheng Lee** is a PhD student at the University of Texas at Austin advised by Dr. Joydeep Ghosh. His research work focuses on applications of data mining and machine learning algorithms in biology and medicine. He received his B.S. degrees in computer science and electrical engineering and his M.S. in electrical engineering from the University of Texas at Dallas. He was also a computational biologist in McDermott Center for Human Growth and Development at the University of Texas Southwestern Medical Center.

**Joydeep Ghosh** received the B.Tech. degree from the Indian Institute of Technology Kanpur in 1983 and the Ph.D. degree from the University of Southern California in 1988. He is currently the Schlumberger Centennial Chair Professor with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, where he has been with the faculty since 1988. He has published more than 250 refereed papers and 50 book chapters, coedited 20 books, and received 14 best paper awards.